

```

start,image<
slip<
i=10
b a50,b50,c50,d50,e50
; time2: 520.43s
; time2: 506.46s = 8m 26.46s    arn ()D1
; time2: 503.88s = 8m 23.88s    V
; time2: last: 201841.61s = 2d 8h 4m 1.61s
; time3: 4742.05s = 1h 19m 2.05s
; time3: last: 2883.63s = 48m 3.63s
; time5: 1206.11s = 20m 6.11s
; time6: 1160.17s = 19m 20.17s
; time7: 1122.83s = 18m 42.83s
;
; Table e1:
;
; For board[p], store the relative address of the first cell in the row in d2:
;
e1:  it p- 0, pa d2      ; 1 1
      it p- 1, pa d2      ; 1 2
      it p- 2, pa d2      ; 1 3
      it p- 3, pa d2      ; 1 4
      it p- 4, pa d2      ; 1 5
      it p- 5, pa d2      ; 1 6
      it p- 6, pa d2      ; 1 7
      it p- 7, pa d2      ; 1 8
      it p- 8, pa d2      ; 1 9
      it p- 0, pa d2      ; 2 1
      it p- 1, pa d2      ; 2 2
      it p- 2, pa d2      ; 2 3
      it p- 3, pa d2      ; 2 4
      it p- 4, pa d2      ; 2 5
      it p- 5, pa d2      ; 2 6
      it p- 6, pa d2      ; 2 7
      it p- 7, pa d2      ; 2 8
      it p- 8, pa d2      ; 2 9
      it p- 0, pa d2      ; 3 1
      it p- 1, pa d2      ; 3 2
      it p- 2, pa d2      ; 3 3
      it p- 3, pa d2      ; 3 4
      it p- 4, pa d2      ; 3 5
      it p- 5, pa d2      ; 3 6
      it p- 6, pa d2      ; 3 7
      it p- 7, pa d2      ; 3 8
      it p- 8, pa d2      ; 3 9
      it p- 0, pa d2      ; 4 1
      it p- 1, pa d2      ; 4 2
      it p- 2, pa d2      ; 4 3
      it p- 3, pa d2      ; 4 4
      it p- 4, pa d2      ; 4 5
      it p- 5, pa d2      ; 4 6
      it p- 6, pa d2      ; 4 7
      it p- 7, pa d2      ; 4 8
      it p- 8, pa d2      ; 4 9
      it p- 0, pa d2      ; 5 1
      it p- 1, pa d2      ; 5 2
      it p- 2, pa d2      ; 5 3
      it p- 3, pa d2      ; 5 4
      it p- 4, pa d2      ; 5 5
      it p- 5, pa d2      ; 5 6
      it p- 6, pa d2      ; 5 7
      it p- 7, pa d2      ; 5 8
      it p- 8, pa d2      ; 5 9
      it p- 0, pa d2      ; 6 1

```

```

it p- 1, pa d2      ; 6 2
it p- 2, pa d2      ; 6 3
it p- 3, pa d2      ; 6 4
it p- 4, pa d2      ; 6 5
it p- 5, pa d2      ; 6 6
it p- 6, pa d2      ; 6 7
it p- 7, pa d2      ; 6 8
it p- 8, pa d2      ; 6 9
it p- 0, pa d2      ; 7 1
it p- 1, pa d2      ; 7 2
it p- 2, pa d2      ; 7 3
it p- 3, pa d2      ; 7 4
it p- 4, pa d2      ; 7 5
it p- 5, pa d2      ; 7 6
it p- 6, pa d2      ; 7 7
it p- 7, pa d2      ; 7 8
it p- 8, pa d2      ; 7 9
it p- 0, pa d2      ; 8 1
it p- 1, pa d2      ; 8 2
it p- 2, pa d2      ; 8 3
it p- 3, pa d2      ; 8 4
it p- 4, pa d2      ; 8 5
it p- 5, pa d2      ; 8 6
it p- 6, pa d2      ; 8 7
it p- 7, pa d2      ; 8 8
it p- 8, pa d2      ; 8 9
it p- 0, pa d2      ; 9 1
it p- 1, pa d2      ; 9 2
it p- 2, pa d2      ; 9 3
it p- 3, pa d2      ; 9 4
it p- 4, pa d2      ; 9 5
it p- 5, pa d2      ; 9 6
it p- 6, pa d2      ; 9 7
it p- 7, pa d2      ; 9 8
it p- 8, pa d2      ; 9 9
;
; Table e2:
;
; For board[p], store the relative address of the first cell in the column in d2:
;
e2:  it p- 0, pa d2      ; 1 1
      it p- 0, pa d2      ; 1 2
      it p- 0, pa d2      ; 1 3
      it p- 0, pa d2      ; 1 4
      it p- 0, pa d2      ; 1 5
      it p- 0, pa d2      ; 1 6
      it p- 0, pa d2      ; 1 7
      it p- 0, pa d2      ; 1 8
      it p- 0, pa d2      ; 1 9
      it p- 9, pa d2      ; 2 1
      it p- 9, pa d2      ; 2 2
      it p- 9, pa d2      ; 2 3
      it p- 9, pa d2      ; 2 4
      it p- 9, pa d2      ; 2 5
      it p- 9, pa d2      ; 2 6
      it p- 9, pa d2      ; 2 7
      it p- 9, pa d2      ; 2 8
      it p- 9, pa d2      ; 2 9
      it p-18, pa d2      ; 3 1
      it p-18, pa d2      ; 3 2
      it p-18, pa d2      ; 3 3
      it p-18, pa d2      ; 3 4
      it p-18, pa d2      ; 3 5
      it p-18, pa d2      ; 3 6

```

```

it p-18, pa d2      ; 3 7
it p-18, pa d2      ; 3 8
it p-18, pa d2      ; 3 9
it p-27, pa d2      ; 4 1
it p-27, pa d2      ; 4 2
it p-27, pa d2      ; 4 3
it p-27, pa d2      ; 4 4
it p-27, pa d2      ; 4 5
it p-27, pa d2      ; 4 6
it p-27, pa d2      ; 4 7
it p-27, pa d2      ; 4 8
it p-27, pa d2      ; 4 9
it p-36, pa d2      ; 5 1
it p-36, pa d2      ; 5 2
it p-36, pa d2      ; 5 3
it p-36, pa d2      ; 5 4
it p-36, pa d2      ; 5 5
it p-36, pa d2      ; 5 6
it p-36, pa d2      ; 5 7
it p-36, pa d2      ; 5 8
it p-36, pa d2      ; 5 9
it p-45, pa d2      ; 6 1
it p-45, pa d2      ; 6 2
it p-45, pa d2      ; 6 3
it p-45, pa d2      ; 6 4
it p-45, pa d2      ; 6 5
it p-45, pa d2      ; 6 6
it p-45, pa d2      ; 6 7
it p-45, pa d2      ; 6 8
it p-45, pa d2      ; 6 9
it p-54, pa d2      ; 7 1
it p-54, pa d2      ; 7 2
it p-54, pa d2      ; 7 3
it p-54, pa d2      ; 7 4
it p-54, pa d2      ; 7 5
it p-54, pa d2      ; 7 6
it p-54, pa d2      ; 7 7
it p-54, pa d2      ; 7 8
it p-54, pa d2      ; 7 9
it p-63, pa d2      ; 8 1
it p-63, pa d2      ; 8 2
it p-63, pa d2      ; 8 3
it p-63, pa d2      ; 8 4
it p-63, pa d2      ; 8 5
it p-63, pa d2      ; 8 6
it p-63, pa d2      ; 8 7
it p-63, pa d2      ; 8 8
it p-63, pa d2      ; 8 9
it p-72, pa d2      ; 9 1
it p-72, pa d2      ; 9 2
it p-72, pa d2      ; 9 3
it p-72, pa d2      ; 9 4
it p-72, pa d2      ; 9 5
it p-72, pa d2      ; 9 6
it p-72, pa d2      ; 9 7
it p-72, pa d2      ; 9 8
it p-72, pa d2      ; 9 9
;
; Table e3:
;
; For board[p], store the relative address of the first cell in the submatrix in d2:
;
e3:  it p- 0, pa d2      ; 1 1
      it p- 1, pa d2      ; 1 2

```

it p- 2, pa d2	; 1 3
it p- 0, pa d2	; 1 4
it p- 1, pa d2	; 1 5
it p- 2, pa d2	; 1 6
it p- 0, pa d2	; 1 7
it p- 1, pa d2	; 1 8
it p- 2, pa d2	; 1 9
it p- 9, pa d2	; 2 1
it p-10, pa d2	; 2 2
it p-11, pa d2	; 2 3
it p- 9, pa d2	; 2 4
it p-10, pa d2	; 2 5
it p-11, pa d2	; 2 6
it p- 9, pa d2	; 2 7
it p-10, pa d2	; 2 8
it p-11, pa d2	; 2 9
it p-18, pa d2	; 3 1
it p-19, pa d2	; 3 2
it p-20, pa d2	; 3 3
it p-18, pa d2	; 3 4
it p-19, pa d2	; 3 5
it p-20, pa d2	; 3 6
it p-18, pa d2	; 3 7
it p-19, pa d2	; 3 8
it p-20, pa d2	; 3 9
it p- 0, pa d2	; 4 1
it p- 1, pa d2	; 4 2
it p- 2, pa d2	; 4 3
it p- 0, pa d2	; 4 4
it p- 1, pa d2	; 4 5
it p- 2, pa d2	; 4 6
it p- 0, pa d2	; 4 7
it p- 1, pa d2	; 4 8
it p- 2, pa d2	; 4 9
it p- 9, pa d2	; 5 1
it p-10, pa d2	; 5 2
it p-11, pa d2	; 5 3
it p- 9, pa d2	; 5 4
it p-10, pa d2	; 5 5
it p-11, pa d2	; 5 6
it p- 9, pa d2	; 5 7
it p-10, pa d2	; 5 8
it p-11, pa d2	; 5 9
it p-18, pa d2	; 6 1
it p-19, pa d2	; 6 2
it p-20, pa d2	; 6 3
it p-18, pa d2	; 6 4
it p-19, pa d2	; 6 5
it p-20, pa d2	; 6 6
it p-18, pa d2	; 6 7
it p-19, pa d2	; 6 8
it p-20, pa d2	; 6 9
it p- 0, pa d2	; 7 1
it p- 1, pa d2	; 7 2
it p- 2, pa d2	; 7 3
it p- 0, pa d2	; 7 4
it p- 1, pa d2	; 7 5
it p- 2, pa d2	; 7 6
it p- 0, pa d2	; 7 7
it p- 1, pa d2	; 7 8
it p- 2, pa d2	; 7 9
it p- 9, pa d2	; 8 1
it p-10, pa d2	; 8 2
it p-11, pa d2	; 8 3

```

it p- 9, pa d2      ; 8 4
it p-10, pa d2      ; 8 5
it p-11, pa d2      ; 8 6
it p- 9, pa d2      ; 8 7
it p-10, pa d2      ; 8 8
it p-11, pa d2      ; 8 9
it p-18, pa d2      ; 9 1
it p-19, pa d2      ; 9 2
it p-20, pa d2      ; 9 3
it p-18, pa d2      ; 9 4
it p-19, pa d2      ; 9 5
it p-20, pa d2      ; 9 6
it p-18, pa d2      ; 9 7
it p-19, pa d2      ; 9 8
it p-20, pa d2      ; 9 9
;
;   stack:
;
e4:   i=i+162
;
;   board:
;
;   Last cell in row f-marked (for print), very last cell ,-marked as well
;
e5:   qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qqf 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qqf 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qqf 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qq 0
      qqf 0
      qq 0
      qq 0
      qq 0
      qq 0

```

```

qq 0
qq 0
qq 0
qq 0
qqf 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qqf 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qqf 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qqf 0
qq 0
qq 0
qq 0
qq 0
qq 0
qq 0
qqf 0,
e6:  qq                                ; save s
;
;    possible[0:80]
;
;  For each cell, bit n (1-9) is set if n is possible for this cell
;
e7:  qq                                ; possible
      i=i+80
e8:  qq  0
      qq  1
      qq  2
      qq  3
      qq  4
      qq  5
      qq  6
      qq  7
      qqf 8
      qq  0
      qq  9
      qq 18
      qq 27
      qq 36
      qq 45

```

```

qq 54
qq 63
qqf72
qq 0
qq 1
qq 2
qq 9
qq 10
qq 11
qq 18
qq 19
qqf20,
d3: qq 511.9+1023.19+1023.29+1023.39
d4: qq ; m
d5: qq ; best p
d7: qq ; best m
d8: qq ; m2
;
; test print
;
; write({nddd},p+1,Raddr);
;
d9: gr d10 , gs d11
sy 64
arn p+1 D
ck -30
qq b8 , hs b10 ; write({nddd}, R);
arn d10 , ck -30
qq b8 , hs b10 ; write({nddd}, R);
d11: ps 0 ; s:=save s
hr s1
d10: qq
;
; Utility sequence for generating possible[p]:
;
; ud pe0, qq
; arn d2, ga d21
; qq offset, hs d20
;
hr s1 ; end rowcolsub;
;
a0: ps 0 , pp 0 ; start: s:=0; p:=0;
; zq
vy 16 ; select(16);
;
; Read sudoku:
;
; Only read digits, ignore rest of the characters incl. case shifts:
;
pa c3 t e5-1 ; c3:=addr(board[-1]);
c2: lyn D ; again: c2:=Raddr:=lyn;
ca 11 , hsf 2 ; if Raddr=11 then exit;
ca 16 , hvn c3 ; if Raddr=16 then begin R:=0; goto digit end;
is (c2)
bs s502 t 502 ; if char>0 ^ char<10 then
hv c3 ; goto digit;
hv c2 ; goto again:
c3: gr 0 t 1 ; digit: board[count++]:=R;
arn(c3) IPC ; R:=board[count]; PC:=marks
hv c2 NPA ; if -,PA then goto again;
hs b0 ; print;
zl ; reset clock;
a1: grn d5 ; best p:=0;
pt a5 t 10 ; best n:=10;

```

```

        grn d7                ; best m:=0;
        pp 0
a2:   grn pe7                ; for p:=0 step 1 until 80 do begin possible[p]:=0;
        arn pe5              IPC ; R:=board[p]; PC:=marks;
        hv a8                NZ  ; if board[p]=0 then begin
        arn 511 D
        gr d4                ; m:=511.9;
        pa d30 t e8-1
        pa d32 t e1-81
d1:   it 81
d32:  ud pe1-81, qq         ; d2:=rows[p]
d30:  arn e8-1 t 1         IQC ; get offset
        ga d31
        arn d3              ; R:=1 0 39 m;
d31:  is 0                 ; s:=offset;
d2:   is s0                ; s:=s+result of ud;
        ns (se5)            ; s:=-board[s];
        ck s                ; R:=R shift s;
        mb d4               ; R:=R ^ m;
        gr d4               ; m:=R;
        hv d30              NQB ;
        hv d1               NQA
        pa a5               ; n:=0;
        arn d4 , gr pe7     ; R:=m; possible[p]:=m;
a4:   hv a5                 LZ  ; if R=0 then goto a5
        qq (a5) t 1         ; n:=n+1;
        nk 0 D              ; normalize;
        mb 767 D           ; remove bit 1
        hv a4
a5:   it 0 , bs 0          ; if n<best n then
        hv a6
a8:   pp p1
        hv a2              NPA ;
        hv a7              ; done p loop
a6:   it (a5) , pt a5      ; best n:=n;
        gp d5              ; best p:=p;
        pm d4 , gm d7      ; best m:=m;
        hv a8
a7:   arn a5 , ck 10
        mb 511 D           ; Raddr:=best n
        hv a9              NZ  ; if best n=0 then comment dead end;
a12:  ps s-2              ; destack: s:=s-2;
        pp (se4)           ; p:=stack[s];
        grn pe5            ; board[p]:=0;
;     hs d9               ; test print
        hv a10             ; goto try next;
a9:   ca 10 , hv a11       ; if best n=10 then goto print;
        ca 1 , hv a20      ; if best n=1 then goto better;
;
;     Find row/col/submatrix with only one candidate:
;
;     Use possible[p] to find the only candidate:
;
        pp 0
a25:  arn pe5              IPC ; loop p: R:=board[p]
        hv a21             NZ  ; if board[p]=0 then begin
        arn pe7 , gr d4    ; m:=possible[p];
a22:  arn d4              ; next m: R:=m;
        hv a21             LZ  ; if m=0 then goto next p;
        nk a23             ; normalize to find first bit set
        mb 767 D           ; remove left bit
        ck (a23)           ; shift back
        gr d4              ; m:=R;
        arn 256 D          ; R:=2 1 38 0;

```



```

a23: ck  0                      ; R:=R shift -j;
      gr  d8                      ; m2:=R;
      pa  d40    t e8-1
      pa  d42    t e1-81
d43: it  81
d42: ud  pe1-81, qq              ; d2:=rows[p]
      arn d2    , ga  d21        ; d21:=d2;
      pa  d23                      ; k:=0;
d40: arn e8-1  t  1    IQC ; get offset
      ga  d41
d41: is  0                      ; s:=offset;
d21: is s0                      ; s:=s+d21
      arn se7                      ; R:=possible[d21+offset]
      mb  d8                      ; R:=R^m2;
d23: qq  0    t  1    NZ        ; if R≠0 then k:=k+1;
      hv  d40    NQB ;
      arn d23                      ; R:=k;
      nc  1      , hv  a26        ; if k=1 then begin
a27: gp  d5                      ; this: best p:=p;
      arn d8
      gr  d7      , hv  a20        ; best m:=m2; goto better; end;
a26: hv  d43    NQA
      hv  a22                      ; goto next m;
a21: pp  p1                      ; next p: p:=p+1;
      hv  a25    NPA ; if -,last goto loop p;
a20: it  (d5)    , pa  se4        ; better: stack[s]:=best p;
      pm  d7      , gm  sle4      ; stack[s+1]:=best m;
      hv  a10
;
;  print
;
a11: zl                      ; RF:=clock count;
      sy  64                      ; writecr;
      qq  b9      , hs  b11        ; write({nddddddddddd.dd}, RF);
      hs  b0                      ; print;
      hv  a0                      ; goto start;
;
a10: pp  (se4)                  ; p:=stack[s];
      arn sle4                  ; R:=stack[s+1];
      hv  a12    LZ            ; if R=0 then goto destack;
      nk  a13                      ; normalize
      mb  767    D              ; remove leftmost bit
      ck  (a13)                  ; shift back
      gr  sle4                  ; stack[s+1]:=leftover;
a13: srn 0    D -1              ; Raddr:=digit;
      gr  pe5                      ; board[p]:=digit;
;  hs  d9                      ; test print
      ps  s2                      ; s:=s+2;
      hv  a1                      ; goto a1;
;
;  procedure print;
;
b0:  sy  64                      ; writecr;
      pa  b1      t e5-1        ; set address
b1:  arn 0      t  1    IPC      ; nextdigit: R:=board[count++]; PC:=marks;
      ga  b2                      ; b2:=Raddr;
      sy  0                      ; writechar(0);
      ca  0      , it  16        ; if Raddr=0 then writechar(16) else
b2:  sy  0                      ; writechar(Raddr);
      sy  64    LPB            ; if count mod 9=0 then writecr;
      hv  b1    NPA            ; if count≠81 then goto nextdigit;
      hr  s1
; Layout for print sequence:
; qq <b>.3+<h>.7+<f1>.9+<d>.13+<n>.14+<bE>.17+<f2>.19+<g1>.23+<g2>.27+<g3>.31+<g4>.35

```

```

; 0 ≤ b ≤ 15    No. of significant digits
; 0 ≤ h ≤ 15    Digits before .
; 0 ≤ d ≤ 15    No. of decimals
; 0 ≤ f1 ≤ 3     sign
; 0 ≤ n ≤ 1     0 in front
; 0 ≤ bE ≤ 7     digits in exponent
; 0 ≤ f2 ≤ 3     sign on exponent
; 0 ≤ gx ≤ 15    grouping
b8:  qq  4.3+ 4.7+0.13+1.14+ 4.23 ;{nddd}
b9:  qq 14.3+12.7+2.13+1.14+12.23+ 2.27 ;{nddddddddddd.dd}

```

[The following taken from print in HELP 3]

[NUMBER PRINT ROUTINE

page 1]

```

b a51                                ;
;
a42: qq  0      , ck  0      ; entry address part 0≤x<1024
a50: qq  570    , t1  -30    ; entry address part -512≤x≤511
b10: pm  28     DV          ; entry integer
      pm -11     D          ; entry fractional
b11: pa  ra11 X      t485    ; entry real; numberpart:=true;
      gs  ra12 , gp  ra13    ; save p; save s;
      ga  ra14 , gm ra23    ; store exp2; store numberpart;
      pm  (s) , arn  s      ;
      pm  (s1)              NA ; get layout address
a14: psn  0      X          ; s:=exp2;
      ps  s11 , cl  -20     ; s:=exp2+11;
      tk  14 , ga  ra17    ; unpack layout: b
      tk  10 , pa  ra16    ; exp10:=0;
a40: it  0      , pa  ra4    ; expprinting:
      pp  256 , ck  -6      ; bE:=256;
      ga  ra18 , tk  10     ; h
      ck  -8 , ga  ra19    ; f1
      tk  10 , ck  -6      ;
      ga  ra20 , tk  11     ; d
      tk  -20 , gt  ra21    ; -n
      tk  20 , ck  -1      ;
      ga  ra22 , tk  -6     ; bE+f2
      ca  1 , pp  10       ; if bE=1 then minexp:=10;
      ca  2 , pp  100      ; if bE=2 then minexp:=100;
      tln 34 , ar  ra43    ; group picture
      gr  ra8 , snn ra23    ; R:=-abs(numberpart);
      pa  ra4 V t-15 LZ ; if numberpart=0 then H:=-15
a49: nk  ra46 XV          ; reconversion: else x:=numberpart×2exp2;
      gr  ra26 , hv  ra31    ; goto if numberpart=0 then L3
      bs  s-11 , hv  ra47    ; else if s>0 then conversion1
; else conversion2;
a34: hv  ra48 , sr  ra26    ; round x2: R:=rounded x;
a17: pa  0      XVD t11 NO ; if -,overflow then b:=11
      mt  ra7 , hv  ra49    ; else goto reconversion;
      bs  (ra4) , ntn (ra4) ; if H>0 then begin R:=0;
a21: qq (ra18)          ; h:=h-H end else h:=h-n;
a19: pp  0      , gt  ra13    ; p:=f1; a13incr:=R;
      bs  p509 , hv  ra36    ; if p≠3 then goto count h
a11: arn 485 D t-485      ; exppart:=-,numberpart; numberpart:=false;
      hs  ra5              LT ; if exppart then write 10
      bs  (ra24) , arn ra23 ; if b1>0^x<0 then
[21.9.1967 (10) PRINT, PAIR page 18]

```

[NUMBER PRINT ROUTINE

page 2]

```

arn -480 DV      NT ; R:=-
arn  32  DV      ;  else R:=+

```

```

        bs p510 , ck 10 ; if R<0^p<2 then R=small
        hs ra37 , pp 3 ; write sign
a36: ;
a18: bt 0 , t-1 ; count h:
        hsn ra2 , hv ra36 ; write space before digits
        bs p509 , hv ra11 ;
a13: pp 0 , ncn 0 ; restore p;
        hsn ra1 , qq ;
        bt (ra4) , t-1 ; count H
        hsn ra , hv r-1 ; write digits before point
        arn ra38 , bs (ra20) ; if d>0 then
        hs ra5 , it -1 ; write point;
a20: bt 0 , hh ra39 ; count d, write decimals
a12: ps 0 , xrn ; restore s; M:=0;
        bs (ra24) , pm ra16 ; if b2>0 then M:=exp10;
        can (ra22) , hr s1 ; EXIT
        gm ra23 , srn ra9 ;
        pm (ra22) DX ;
        ps 9 , hh ra40 ; goto expprinting;
a37: ca p , hh s ; if p=0^Raddr=0 then go back
        gr ra26 , arn ra9 ; sign not counted in group
        ac ra8 , arn ra26 ;
a5: bs (ra) , hv ra1 ; if b1>0 then write digit
        mb ra44 , ga ra6 ;
a39: hvn ra10 , arn ra3 ; else if digit≠0 then write digit
a4: it 0 , t1 ; count H else write 0
        bs 0 , hvn ra5 ; if H<0 then write 0
a: bt 0 , t-1 ; count b1
        mln ra3 , tk 30 ; next digit in R
a1: ar 16 D LZ ; zero instead of space
a2: ga ra6 , bs (ra50) ; if actual case=upper then
        mt ra7 , it 510 ; R:=-R;
        sy (ra50) , t-510 LT ; write case;
a10: qq (ra42) , t1 ; count printed;
a51: arn ra8 , sr ra9 ; actual group:= actual group-1;
        gr ra8 , nc -273 ; if -,group full then write out
a6: sy 0 , hhn s ; else if out>58 then begin
        tk 4 , it 58 ; actual group:= next group;
        bs (ra6) , hv rla51 ; write out end
[21.9.1967 (10) PRINT, PAIR page 19]

```

[NUMBER PRINT ROUTINE

page 3]

```

a44: sy -256 , ud ra10 ; else begin writespace; count printed;
        ; actual group:=next group-1; write out end;
a41: hh ra51 , it 1 ; conversion:
a28: qq (ra4) , t-1 ; begin comment (count H)
a25: ps s-3 , nk r1 ; by multiplication by 213/10 or 10/214
        ps s0 , gr ra26 ; x is converted to form
        pm ra26 , bs (ra46) ; x=x2×101H where x>x2≥.1
a47: mkn ra27 , hh ra41 ; conversion 1:
a46: tk s , gr ra26 ;
a48: ps s7 , ar ra29 ; conversion 2:
        pm -1 DV LT ;
        mkn ra30 , hv ra28 ; end conversion;
        ps (ra4) , can (ra22) ; s:=H; if bE=0^f2=0 then begin
        bs s-15 , it 64 ; if s>15 then bE=1 end;
a22: ca 0 , hv ra31 ; if bE^f2=0 then goto L3;
        arn ra17 , sr ra20 ; R:=b-d;
        ga ra32 , sr ra18 ; a32:=b-d; Raddr:=b-d-h-1;
a7: mb -1 DX LT ; M:=if b-d-h-1<0 then b-d-h-1 else -1;
a38: xr 315 , it (ra4) ; L1: R:=M;
a32: bs 0 , hs ra33 ; if b-d>H then change exp10

```

```

        mt  ra7    , it  (ra18) ; L2: R:=-R;
        bs  (ra4)  , hs  ra33 ;   if H>b then change exp10
a31:  arn ra20    , ga  ra   ; L3: b1:=d;
a45:  ar  ra4    , ga  ra24 ; L4: b2:=H+d;
a24:  bs  0      , ga  ra   ;   if b2>0 then b1:=b2;
        it  (ra17) , bs  (ra24);   if b<H-d then
        arn ra17  , hh  ra45 ;     b2:=b; goto L4;
        arn 256 D      NT ;     if b2>0 then R:=.5;
a35:  ps  (ra24) , pm  ra21 ; rounding: s:=b2;
        bs  s511  , hh  ra34 ;   if s<0 then goto round x2;
a16:  xr  p0      , mln ra29 ;     R:=RX.1;
        ps  s-1   , hh  ra35 ;     s:=s-1; goto rounding;
        ; change exp10:
a33:  ac  ra16    , bs  (ra16) ;   exp10:=exp10+R; if exp10>minexp then
        sc  ra4    , hh  s-1  ;     begin H:=H-R; goto L1 vL2 end;
        sc  ra16   , hv  ra31 ;     else exp10:=exp10-R; goto L3;
        ;
a3:   10          ;
a9:   qq  1        ;
a27:  can  s409   , cm  (r-410);   0.8
a29:  vy   p51    , mln (204) ;    0.1
a30:  qq   320    ;    10/16
a43:  qq  -17.5+1.25-1.39 ;
a8:   qq
a26:  qq
a23:  qq
e      ; end write;
ea0    ; goto start;
8,0,0,0,0,0,0,0,0,0,
0,0,3,6,0,0,0,0,0,0,
0,7,0,0,9,0,2,0,0,0,
0,5,0,0,0,7,0,0,0,0,
0,0,0,0,4,5,7,0,0,0,
0,0,0,1,0,0,0,3,0,0,
0,0,1,0,0,0,0,6,8,0,
0,0,8,5,0,0,0,1,0,0,
0,9,0,0,0,0,4,0,0,0,

003020600
900305001
001806400
008102900
700000008
006708200
002609500
800203009
005010300

200080300
060070084
030500209
000105408
000000000
402706000
301007040
720040060
004010003

000000907
000420180
000705026
100904000
050000040
000507009

```

920108000
034059000
507000000

030050040
008010500
460000012
070502080
000603000
040109030
250000098
001020600
080060020

020810740
700003100
090002805
009040087
400208003
160030200
302700060
005600008
076051090

100920000
524010000
000000070
050008102
000000000
402700090
060000000
000030945
000071006

043080250
600000000
000001094
900004070
000608000
010200003
820500000
000000005
034090710

480006902
002008001
900370060
840010200
003704100
001060049
020085007
700900600
609200018

000900002
050123400
030000160
908000000
070000090
000000205
091000050
007439020
400007000

001900003
900700160
030005007
050000009
004302600
200000070
600100030
042007006
500006800

000125400
008400000
420800000
030000095
060902010
510000060
000003049
000007200
001298000

062340750
100005600
570000040
000094800
400000006
005830000
030000091
006400007
059083260

300000000
005009000
200504000
020000700
160000058
704310600
000890100
000067080
000005437

630000000
000500008
005674000
000020000
003401020
000000345
000007004
080300902
947100080

000020040
008035000
000070602
031046970
200000000
000501203
049000730
000000010
800004000

361025900
080960010
400000057
008000471

000603000
259000800
740000005
020018060
005470329

050807020
600010090
702540006
070020301
504000908
103080070
900076205
060090003
080103040

080005000
000003457
000070809
060400903
007010500
408007020
901020000
842300000
000100080

003502900
000040000
106000305
900251008
070408030
800763001
308000104
000020000
005104800

000000000
009805100
051907420
290401065
000000000
140508093
026709580
005103600
000000000

020030090
000907000
900208005
004806500
607000208
003102900
800605007
000309000
030020050

005000006
070009020
000500107
804150000
000803000
000092805
907006000
030400010

200000600

040000050
001943600
009000300
600050002
103000506
800020007
005000200
002436700
030000040

004000000
000030002
390700080
400009001
209801307
600200008
010008053
900040000
000000800

360020089
000361000
000000000
803000602
400603007
607000108
000000000
000418000
970030014

500400060
009000800
640020000
000001008
208000501
700500000
000090084
003000600
060003002

007256400
400000005
010030060
000508000
008060200
000107000
030070090
200000004
006312700

000000000
079050180
800000007
007306800
450708096
003502700
700000005
016030420
000000000

030000080
009000500

007509200
700105008
020090030
900402001
004207100
002000800
070000090

200170603
050000100
000006079
000040700
000801000
009050000
310400000
005000060
906037002

000000080
800701040
040020030
374000900
000030000
005000321
010060050
050802006
080000000

000000085
000210009
960080100
500800016
000000000
890006007
009070052
300054000
480000000

608070502
050608070
002000300
500090006
040302050
800050003
005000200
010704090
409060701

050010040
107000602
000905000
208030501
040070020
901080406
000401000
304000709
020060010

053000790
009753400
100000002
090080010
000907000
080030070

500000003
007641200
061000940

006080300
049070250
000405000
600317004
007000800
100826009
000702000
075040190
003090600

005080700
700204005
320000084
060105040
008000500
070803010
450000091
600508007
003010600

000900800
128006400
070800060
800430007
500000009
600079008
090004010
003600284
001007000

000080000
270000054
095000810
009806400
020403060
006905100
017000620
460000038
000090000

000602000
400050001
085010620
038206710
000000000
019407350
026040530
900020007
000809000

000900002
050123400
030000160
908000000
070000090
000000205
091000050
007439020
400007000

380000000
000400785
009020300
060090000
800302009
000040070
001070500
495006000
000000092

000158000
002060800
030000040
027030510
000000000
046080790
050000080
004070100
000325000

010500200
900001000
002008030
500030007
008000500
600080004
040100700
000700006
003004050

080000040
000469000
400000007
005904600
070608030
008502100
900000005
000781000
060000010

904200007
010000000
000706500
000800090
020904060
040002000
001607000
000000030
300005702

000700800
006000031
040002000
024070000
010030080
000060290
000800070
860000500
002006000

001007090
590080001
030000080
000005800

[illegible]

000900006000000008004760003000000000850000000040002030790000000001005090000008020
003002004080000950000000080000070000001050406000080001000001000050000000002400003
01040000050000006000000070025900000000900004006000800000056000000002001008700009
002000080000730000000000006900400020000000040700500000500020001004008300060000700
210000000000604003000000000080010700003005000000080100005300009000200060070000800
900006010000000003000090000014000000006000500000870090008300004000000008230001000
08000004090003000000700216000090007005000020030008000000000000002006700000001009
001300000020000900000007000100000000060540080000090070900103000080000065000000020
064000000000000732000000050000070004207005000000900008000050000003000190000068000
090000200700500000000080000000700069102000800000000000004020000000060050360009070
0005000700000000020010040000000070103002906000500000000040013000000000400080700090
040000000000650030009020080086003000000000500000400020000060400300000001200009003
000000230048000000005020009000000081006000000500400003000906500300800004010000000